Field Results of the Control, Navigation, and Mapping Systems of a Hovering AUV

Nathaniel Fairfield, Dominic Jonak, George Kantor, David Wettergreen
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213 USA
Email: {than, dom, kantor, dsw}@cmu.edu

Abstract

This paper describes the evolution of the control, navigation, and mapping capabilities of a hovering autonomous underwater vehicle (AUV) designed to explore flooded cenotes in Mexico as part of the DEPTHX project. The vehicle is equipped with a suite of navigation sensors that allow it to localize itself and and create maps of the complex 3D environments in which it operates. It is passively stabilized in the roll and pitch directions and equipped with six thrusters that allow it to directly actuate each of the remaining degrees of freedom (x, y, z, and heading). We describe the control system of the vehicle, which provides open-water and near-wall manuevering capabilities. Near-wall behaviors are used in support of science operations such as visual wall survey and precise placement of a core sampling tool. We also present the vehicle's navigation system, which combines dead-reckoning, sonar-based localization, and simultaneous localization and mapping (SLAM). We demonstrate the performance of these systems using experimental results from multiple test environments, including a test tank, a flooded limestone quarry, and the La Pilita cenote. Trajectory plots comparing desired and actual motions will be used to demonstrate the various vehicle controllers. Data from repeated localization experiments will be used to provide statistically meaningful measures of the accuracy of the dead-reckoning, localization, and SLAM filters.

Figure 1: The DEPTHX AUV deployed in Poza La Pilita, with the solid core sampling arm extended.

1 Introduction

The DEPTHX (DEep Phreatic THermal eXplorer) project is a three-year NASA-funded effort whose primary objective is to use an autonomous vehicle to explore and characterize the unique biology of the Zacatón cenote. Zacatón, the world's deepest known limestone sinkhole, is a water-filled cavern that is at least 300 meters deep. The depths of Zacatón are geothermally heated with a high sulfur content and a lack of sunlight or dissolved oxygen, making this an ideal place to search for exotic microbial life [Gary, 2002]. The robotic exploration and search for microbial life in Zacatón is an analog mission for the

search for life in the liquid water ocean beneath the frozen surface of Europa.

The DEPTHX robot (Figure 1) is a hovering autonomous underwater vehicle (AUV) designed to explore flooded caverns and tunnels while building 3D maps, collecting environmental data, and obtaining samples from the water column and cavern walls. To accomplish these tasks, the vehicle is equipped with a Doppler velocity logger (DVL), a ring laser gyro-based inertial navigation system (INS), a depth sensor, and an array of 56 narrow beam sonar transducers.

The paper is organized as follows: related research activities in AUV control and the use of AUVs for science and mapping are briefly discussed in Section 2. Then we provide a description of the instrumentation of the DEPTHX vehicle in Section 3. An overview of the vehicle control system is provided in Section 4. Likewise, the navigation and mapping system is described in Section 5. Experimental results are presented in Section 6, followed by some conclusions and a discussion of future work in Section 7.

2 Related Work

There are a variety of techniques employed for determining the position of underwater vehicles – unfortunately GPS does not work underwater – and they can be divided into those that utilize emplaced infrastructure and those that do not. (See Leonard et al. [1998] for a survey.)

When accurate position is needed underwater, many AUVs and human-driven remotely operated vehicles (ROVs) rely on a surveyed array of acoustic beacons, known as a long base-line (LBL) array. Acoustic beacons provide a fixed frame of reference for positioning the vehicle [Whitcomb, 2000].

Over large distances, or in underwater caves and tunnels, the performance of LBL systems is unknown due to signal attenuation, reverberation and multipath. Without a fixed LBL infrastructure, an AUV uses a combination of depth sensors, inertial sensors, and Doppler velocity sensors to compute a dead reckoned estimate of its position while at depth. With high accuracy attitude and depth sensors the uncertainty in the AUV's 3D pose (roll, pitch, yaw, x, y, z) is primarily in x and y. Most underwater navigation systems are based on a Kalman Filter

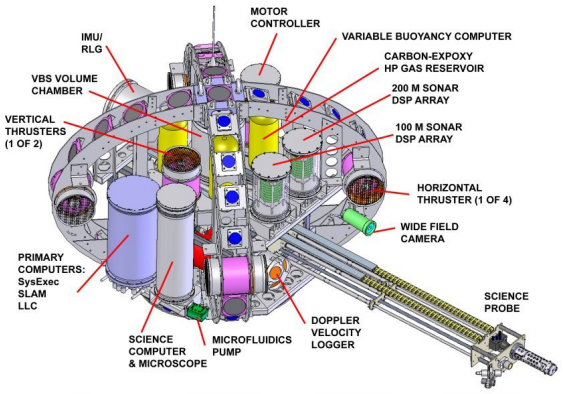

Figure 2: A model of the DEPTHX vehicle structure and components with buoyancy removed for clarity. Eleven pressure vessels house computing, batteries, sensors, and science instruments. Diameter is approximately 2m. ©Stone Aerospace, 2006.

which combines Doppler velocity and inertial measurements. These systems report navigation errors as low as 0.1% of distance traveled [Larsen, 2000], however performance degrades in situations where the DVL is unable to make accurate velocity measurements.

The dead reckoned estimate will accumulate error, and when the drift exceeds what is required for the application, a correction must be made by (re)observing a known reference. A common approach is to surface and obtain position from a GPS. If LBL or surfacing is not an option, the position error can be bounded by simultaneous localization and mapping (SLAM) [Williams et al., 2000] [Dissanayake et al., 2001]. Williams and Mahon [2004] provide an example of near-bottom mapping with sonars and cameras of coral reefs using SLAM. Roman [2005] uses multibeam sonar maps to do SLAM over varied topography.

3 Vehicle Description

The DEPTHX vehicle is a hovering AUV that has been designed for exploration of flooded caverns and tunnels. An oblate spheroid shape approximately 1.5 m in height and 2 m in width (Figures 1 2), the vehicle's dry mass is 1400 kg. Vehicle roll and pitch are stabilized by the sep-

aration of buoyancy and ballast. The vehicle can move directly in the remaining four degrees of freedom (forward, starboard, down, and heading) using six thrusters driven by brushless DC motors. The cruising speed of the vehicle is about 0.2 m per second. The vehicle is powered by two 56-volt Lithium-Ion battery stacks with a total capacity of 6.2 KWh, enough to supply the vehicle during a four hour exploration mission.

The DEPTHX vehicle has a full suite of underwater navigation sensors, including a Honeywell HG2001AC INS, two Paroscientific Digiquartz depth sensors, and an RDI Navigator 600kHz DVL. The specifications for the INS are roll/pitch: 0.2° 2σ, yaw: $0.4^\circ 2\sigma$, for the DVL velocities 0.3 cm/s 1σ, and for the depth sensors 0.01% of full range (10 cm for our 1000m rated sensors). The two depth sensors are zeroed with respect to atmospheric pressure at the start of each day. The DVL is mounted to the front of the vehicle facing forward and tilted down 30 degrees from horizontal, a nonstandard configuration for this instrument. The usual DVL configuration points straight down so that it can achieve lock on the ocean floor. In our application, it is difficult to predict the relative direction to surfaces useful for DVL lock. The top 280 meters of Zacatón is known to be a chimney with a diameter of approximately 80 meters [Fairfield et al., 2005], so the forward-looking configuration should allow the DVL to lock on to one of the vertical walls in most situations. This configuration can cause the DVL to lose bottom lock in more wide-open waters. Loss of bottom lock can also occur at extremely short ranges, or when passing over highly irregular terrain.

For our purposes, the raw roll, pitch, and yaw measurements provided by the IMU and the depth measurements provided by the depth sensors are accurate enough to be considered absolute measurements of those quantities. The task of determining the location of the vehicle is then reduced to the two dimensional problem of estimating its position in the horizontal plane.

For mapping, the vehicle has an array of 56 2° beamwidth sonars that provide a constellation of range measurements around the vehicle. This array is in the shape of three great circles, a configuration that was arrived at after studying the suitability of various sonar geometries for the purposes of SLAM [Fairfield et al., 2005]. The sonars have long ranges (some 100m and others 200m) and the accuracy of the range measurements is fairly high (about 10cm), and were usually fired at about 1 Hz. The low resolution, slow update rate, and sparse point density makes the mapping problem significantly more difficult than it is with ranging sensors like a laser scanner that provide fast, accurate, and high density ranges.

4 Control

The DEPTHX vehicle has a three-level control system that is used to guide the vehicle on its mission. The lowest level, aptly named the low level control system (LLCS), employs velocity feedback from the DVL and IMU in order to generate the thrust necessary to track a desired vehicle frame velocity command. The middle level, named the navigator, issues velocity commands to the LLCS in order to achieve the immediate goal. In this paper, the immediate goal is to drive the vehicle to a specified waypoint, however the navigator also is capable of executing more general behaviors such as wall following and obstacle avoidance. At the highest level is the system executive that, among other things, issues a series waypoint commands to the navigator in order to accomplish the overall mission.

4.1 Low Level Control System

The LLCS performs two basic functions: it uses velocity feedback to convert a vehicle frame velocity command into the vehicle frame thrust needed to track that velocity and it implements a mixing table in order to convert the vehicle frame thrust command into the necessary shaft torque [1] commands to each of the individual thrusters.

Velocity feedback is implemented in four independent loops, one for each of the vehicle's four degrees of freedom. Each loop contains an experimentally tuned PI controller. Note that this structure assumes that the components of vehicle frame velocity are not coupled by the dynamics of the vehicle, an assumption which not true. In particular, the forward and sideways velocity components of the vehicle will be highly coupled when the vehicle simultaneously rotates and moves in the lateral plane.

[1] The relationship between shaft torque and thrust is very nearly linear, and we rely on the DriveBlok^{TM} controller produced by MTS Systems Corp to implement the desired shaft torque on the brushless DC thruster motors.

Hence we can enforce the decoupled assumption by simply avoiding these type of motions, a restriction which is compatible with the slow, deliberate types of missions that the vehicle will undertake. In practice, however, the controller performs well even when such coupling motions are executed.

The thrust mixer maps the vehicle frame thrust command vector $F_v = [F_\omega,\, F_x,\, F_y,\, F_z]^T$ into a vector of n individual thruster commands $T = [\tau_1,\, \tau_2,\, \tau_3,\, \ldots,\, \tau_n]^T$, where n is the number of thrusters. It is implemented as a matrix multiplication, i.e., $T = MF_v$, where M is computed as follows. First, let A be the $6 \times n$ matrix whose ith column is given by

$$A_i = \begin{bmatrix} p_i \times D_i \\ D_i \end{bmatrix},$$

where p_i is a vector describing the location of the ith thruster in the vehicle frame, and D_i is a vector describing the positive thrust direction of the ith thruster in the vehicle frame. Now let B be the $4 \times n$ matrix that is the bottom four rows of A. B is the matrix that maps the individual thruster thrusts T into the resulting vehicle frame thrust vector F_v. Assuming that the rows of B are linearly independent, M can then be found by taking the pseudoinverse of B:

$$M = B^T \left(BB^T\right)^{-1}.$$

Note that in the nominal case, the number of thrusters is $n = 6$. However, this formulation allows the mixing matrix to easily be recomputed in the event of thruster failure.

4.2 Open-water

Waypoint(): Open water waypoint

FlyUpward(): Reactive abort strategy

StaionKeeping(): Holds position and heading

4.3 Near-wall

The vehicle maneuvers relative to a nearby wall using the following behaviors:

ApproachWall(d_0, θ_0): Approach the nearest wall at a standoff distance of d_0 and a relative heading of θ_0 while maintining constant depth. The relative heading is defined to be zero when the vehicle directly faces the wall.

WallFollow($\Delta\ell, \Delta z$): Move rightward $\Delta\ell$ and downward along wall by the distance Δz while maintaining constant standoff distance and relative heading.

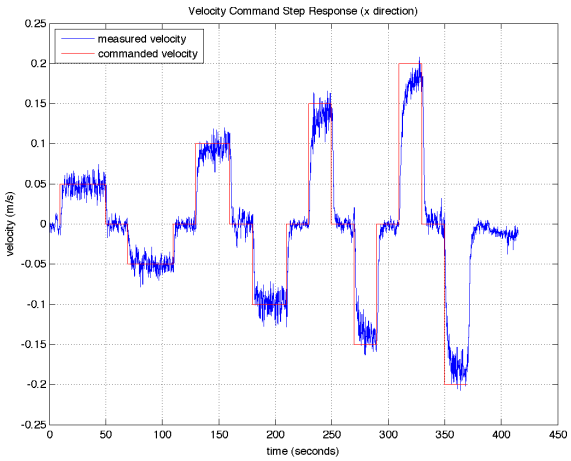

Figure 3: Velocity tracking performance of the LLCS for a commanded velocity pulse. The duration of the leading and falling edge transients is on the order of 4 seconds.

In order to provide these capabilities, we have developed a collection of reactive motion primitives that we call the proxops controller. The proxops controller uses feedback from the mapping sonars to fit a plane to the nearest wall, then issues a velocity command to the vehicle in order to achieve the desired goal. The velocity command is sent to a low-level velocity controller that uses IMU and DVL feedback to determine the thruster inputs necessary to track the desired velocity command.

At the heart of the proxops controller is an inverted dynamics controller derived from the kinematic equations of motion of the vehicle. The equations of motion are derived from the assumption that the wall in front of the vehicle is approximately planar. Under this assumption, the standoff distance d, relative heading θ, and lateral position ℓ can be defined as shown in Figure 4. This geometry can be used to determine the derivatives of these quantities as a function of vehicle velocity:

$$\dot{d} = -v_x + v_y \tan\theta + d\omega \tan\theta,$$

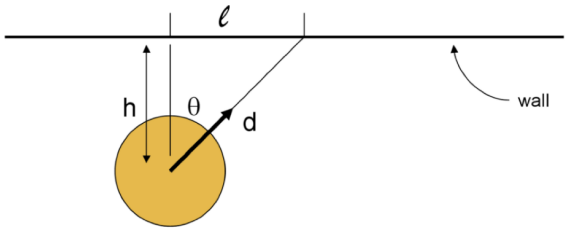

Figure 4: Proximity operations - Definition of control parameters.

controller commands	
d_0	desired wall distance
ℓ_0	desired lateral travel
θ_0	desired wall angle
controller parameters	
γ_d	wall distance convergence rate
γ_ℓ	lateral convergence rate
γ_θ	wall angle convergence rate
k_{dx}	forward damping gain
k_{dy}	lateral damping gain
$k_{d\omega}$	wall angle damping gain

Table 1: Controller notation

$$\dot{\ell} = v_x \sin\theta + v_y \cos\theta + d\omega\cos\theta + \dot{d}\sin\theta,$$
$$\dot{\theta} = \omega$$

where v_x, v_y, and ω are respectively the forward, starboard, and angular velocities of the vehicle.

Using these equations of motion, we determine a proxops control law that maps current wall geometry (d, θ, and ℓ) and vehicle velocity measurements (v_x, v_y, and ω) into desired vehicle velocities ($v_x d$, $v_y d$, and ω_d):

$$v_{xd} = \gamma_d(d - d_0) + v_y\omega\tan\theta + d\omega\tan\theta - k_{dx}v_x,$$
$$v_{yd} = -\frac{\omega d\,(\cos\theta + \sin\theta\tan\theta) - \gamma_\ell(\ell - \ell_0)}{\cos\theta + \omega\sin\theta\tan\theta} - k_{dy}v_y,$$
$$\omega_d = -\gamma_\theta(\theta - \theta_0) - k_{d\omega}\omega,$$

where the controller parameters and commands are defined in Table 1.

This controller assures exponential convergence of the proxops variables d, ℓ, and θ to d_0, ℓ_0, and θ_0, respectively. This can be verified by substituting the desired velocities into the equations of motion. The damping terms

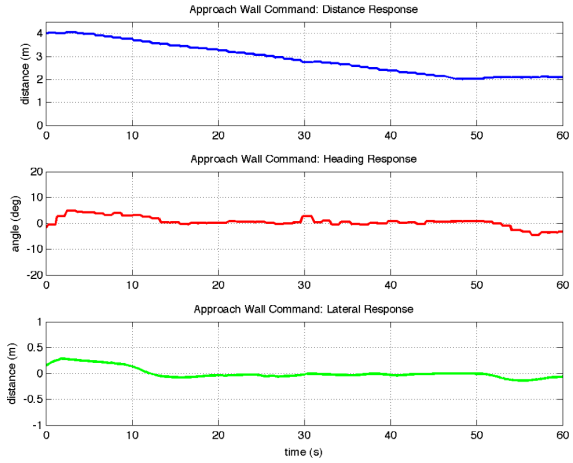

Figure 5: Vehicle response to ApproachWall command.

are hand tuned to account for the fact that the underlying velocity controller is not perfect (e.g., that ω_d is not equal to ω). This control law can be used to achieve all of the desired proxops behaviors by choosing appropriate values for d_0, ℓ_0, and θ_0.

Figures 5 and 6 depict the vehicle response to two types of proxops commands. In both cases these tests were conducted in the same 15 m diameter neutral buoyancy facility used for the SLAM navigation results reported earlier.

In Figure 5 the plots show the vehicle response to an ApproachWall command. The vehicle starts at a distance of 4 m from the wall and is commanded to move to a standoff distance of 2 m at approximately t = 4 seconds. The vehicle reaches its goal at approximately t = 46 seconds. The relative heading and lateral motion are maintained at zero during this maneuver.

In Figure 6 the response to a WallFollowLateral command is shown. Here the vehicle is commanded to follow the wall laterally in the starboard direction for a distance of 5 m. The command is issued at t = 20 seconds, and the goal is reached at about t = 190 seconds. Wall distance and angle are held constant. Note that there is significant noise in the distance and angle measurements due to noisy sonar data (a function of the test tank geometry and the frequency of the tranducers). A noise filter is currently being implemented, but nonetheless the controller performs well in the presence of this noise.

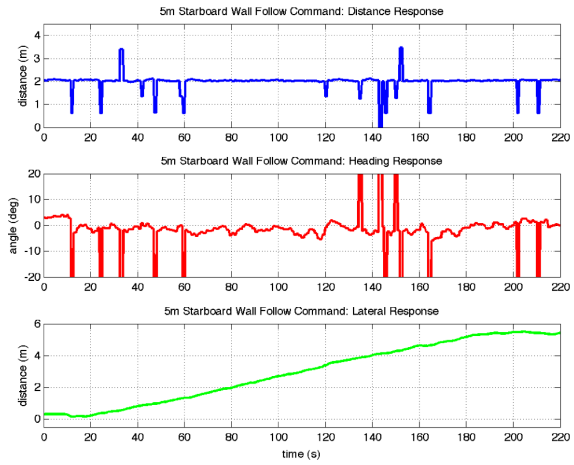

Figure 6: Vehicle response to WallFollow command.

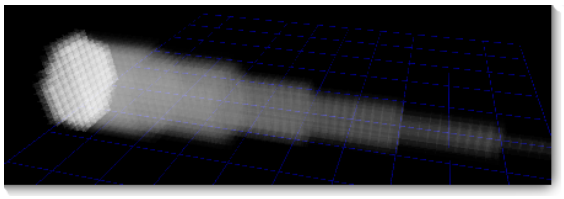

Figure 7: A single 6° sonar beam as represented in an evidence grid.

5 Navigation

5.1 Navigation Architecture

The DEPTHX vehicle used three different navigation methods: dead reckoning, localization with a prior map, and simultaneous localization and mapping. These methods are described below, and are listed in order of complexity. Onboard the vehicle, a simple switching process decided which of the three navigation solutions to use.

5.2 Dead Reckoning

Dead reckoning is the process of integrating the vehicle velocities over time in order to produce a position estimate. In our implementation of dead-reckoning, the algorithm falls back to the IMU velocity measurement when DVL measurements are not available. Unlike the DVL velocities, the IMU velocities drift over time since they are the result of the integration of accelerometers, and this leads to a drift in the position estimate. When the DVL velocities are good, they are used to estimate and correct for the IMU drift using a standard Kalman filter. This allows the IMU to fill in for brief DVL dropouts from noise, switching from water column tracking to wall tracking, and loss of lock when too close to a surface.

Under normal circumstances dead reckoning can provide navigation with a divergence rate of 0.5% of distance traveled. The DEPTHX vehicle can run a 4 hour mission at an average cruising speed of 0.2 m/s, and thus an exploration range of approximately 3000 m. Over this range, the dead-reckoning error would be on the order of 15 m. DVL dropouts significantly degrade this position estimate.

5.3 3D Map

Sonar measurements are noisy and unable to resolve fine features, but over time they do provide information about the environment around the vehicle. In order to combine the individual sonar measurements, the DEPTHX vehicle uses a 3D evidence grid (see Martin and Moravec [1996] for a description of the classic 2D evidence grid). In a 3D evidence grid, space is uniformly discretized into cubic voxel elements.

A measurement model is used to determine how a particular sonar measurement affects the map. We modeled the 56 individual sonars as producing 2-degree cones projecting from the vehicle (see Figure 7). Given a particular range measurement, our model states that voxels within the cone are probably empty and voxels at the end of the cone are probably occupied. A modified Bresenham 3D ray tracing algorithm [Bresenham, 1965] is used to merge these probabilities with the information already contained in the 3D evidence grid, according to a Bayesian update rule.

A major drawback of the 3D evidence grid approach is that the memory required to store them increases as the cube of the size of the map. For reasonable map sizes and resolutions, the memory requirements quickly become intractible, especially considering that the particle filter described below requires hundreds of maps. To cope with this storage and processing problem we use the

Deferred Reference Counting Octree (DCRO) data structure described by Fairfield et al. [2007]. The DRCO is a drop-in replacement for a standard 3D evidence grid that exploits shared regions between particle maps and efficiently represents sparse volumes, yielding a significant performance boost that allows us to represent maps that would not even fit into memory as a uniform array.

5.4 Localization

If the vehicle already has a map of the environment then it can use that map, together with its range sensor measurements, to localize itself. The particle filter localization algorithm has the following steps:

Initialize The particles start with their poses s_0 initialized according to some initial distribution.

Predict The dead-reckoned position innovation u_t is computed using the navigation sensors (IMU, DVL and depth sensor). A new position s_t is predicted for each particle using the vehicle motion model (see Table 3):

$$s_t = h(s_{t-1}, u_t, N).$$

This new distribution of the particles is called the *proposal distribution*.

Weight The weight w for each particle is computed using the measurement model and the sonar range measurements (from $\#_{obs}$ different simultaneous sonar observations):

$$w = \eta \prod_{n=1}^{\#_{obs}} p({}^n z_t | s_t, \Theta)$$

where η is some constant normalizing factor (different than the one used in the expression for the Bayesian filter). In our implementation, the real range measurements z are compared to ray-traced ranges $\hat{z}$ using the particle pose and map. We compare the simulated and real ranges using the measurement model

$$z = g(s_t, u_t, N(0, \sigma_z)),$$

which is assumed to have a normal noise model, so

$$p(z|s, \Theta) = \frac{1}{\sqrt{2\pi\sigma_z^2}} e^{\frac{-(\hat{z}-z)^2}{2\sigma_z^2}}.$$

Substituting into the expression for particle weight and taking the logarithm of both sides shows that maximizing this weight metric is very similar to minimizing the intuitive sum squared error metric:

$$\log w = C - \frac{1}{2\sigma^2} \sum_{i=1}^{\#_{obs}} ({}^i\hat{z} - {}^i z)^2,$$

where $C = \#_{obs} \times \log\left(\sqrt{2\pi\sigma^2}\right)$. An alternative weighting method, called "point correlation" was found to be slightly less informative [Fairfield et al., 2005].

Resample The $O(n)$ algorithm described in [Arulampalam et al., 2002] is used to resample the set of particles according to the weights w such that particles with low weights are likely to be discarded and particles with high weights are likely to be duplicated. The set of particles is now our new estimate of the new vehicle position posterior.

Estimate Generate a position estimate from the particles: when the PF is being used to provide a pose for the rest of the vehicle control software, we usually want to turn the set particles into a single point estimate.

Repeat from Predict

5.5 SLAM

Simultaneous Localization and Mapping (SLAM) is the task of building a map of the environment from sensor data and simultaneously using that map to localize, or recover the robot's actual trajectory. In most cases the robot uses various sensors to measure its own motion and sense its local surroundings. This sensor data is inevitably noisy, and must be appropriately filtered as part of SLAM.

There are a number of different methods which are used to perform SLAM, the most common being based on the well-known Kalman Filter. We present a SLAM method based on a particle filter, another standard approach that is often used when certain requirements of the Kalman Filter formulation (unimodal position distributions, feature detection) cannot be satisfied.

In the most general sense, particle filters sample over the entire state space of the vehicle. As long as this state

$s_t^{(m)}$	vehicle pose of the m-th particle at time t $= (roll, pitch, yaw, x, y, z)^T$
$S_t^{(m)}$	trajectory of m-th particle from time 1 to t $= \{s_1^{(m)}, s_2^{(m)}, \ldots, s_t^{(m)}\}$
z_t	sonar measurements at time t
Z_t	history of measurements from time 1 to t $= \{z_1, z_2, \ldots, z_t\}$
u_t	vehicle dead-reckoned innovation at time t
U_t	history of dead-reckoning from time 1 to t $= \{u_1, u_2, \ldots, u_t\}$
$w_t^{(m)}$	m-th particle weight at time t

Table 2: Particle filter notation.

$N(\mu, \sigma)$	normally distributed noise with mean μ and std dev σ
$h(s_{t-1}, u_t, N(0, \sigma_s))$	vehicle motion model $= p(s_t \mid u_t, s_{t-1})$
$g(s_t, \Theta, N(0, \sigma_z))$	sonar measurement model $= p(z_t \mid s_t, \Theta)$

Table 3: Model notation.

space is fairly low dimensional, as in the case of estimating the 3D position of a vehicle, a few hundred samples will adequately represent the true distribution. But for SLAM, the state space of the vehicle is both the vehicle position *and* the map. This is a very high dimensional space, so we can't possibly apply the basic particle filter: each particle would be a sample from all possible maps and all possible positions within that map.

The goal of SLAM is to estimate the probability distribution at time t over all possible vehicle states s and world maps Θ using all previous sensor measurements Z_t and control commands U_t (for a complete list of notation, see Table 2):

$$p(s, \Theta | Z_t, U_t).$$

This distribution is called the *SLAM posterior*. The recursive Bayesian filter formulation of the SLAM problem is straightforward (see Montemerlo et al. [2002] for a derivation) but the integral is usually computationally intractable to solve in closed form:

$$\overbrace{p(s_t, \Theta | Z_t, U_t)}^{\text{new posterior}} = \eta \times \overbrace{p(z_t | s_t, \Theta)}^{\text{measurement model}} \times \int \underbrace{p(s_t | s_{t-1}, u_t)}_{\text{motion model}} \underbrace{p(s_{t-1}, \Theta | Z_{t-1}, U_{t-1})}_{\text{old posterior}} ds_{t-1},$$

where η is a constant scale factor from Bayes' rule.

The key insight of Murphy [1999] is that the SLAM posterior distribution can be factored into two parts, or marginals: the path distribution and the map distribution. Furthermore, knowing the vehicle's trajectory S_t makes the observations U_t conditionally independent, so that the map sample Θ can be computed in a closed form. The process of factoring a distribution such that one part can be computed analytically is known as Rao-Blackwell factorization [Doucet et al., 2000]. As a result, following Montemerlo et al. [2002] we compute the posterior over *trajectories*, and factor the distribution as

$$p(S_t, \Theta | Z_t, U_t) = p(S_t | Z_t, U_t) p(\Theta | S_t, Z_t).$$

Particle filters are a Monte Carlo approximation to the Bayesian filter. The particle filter maintains a discrete approximation of the SLAM posterior using a (large) set of samples, or *particles*. The m-th instance of the $\#_{par}$ particles represents both a sample pose $S_t^{(m)}$ from the distribution of vehicle trajectories, and the sample map $\Theta^{(m)}$ which results from that trajectory combined with the sensor measurements Z_t. Since we update the particle maps at every time-step, they represent the combination of sensor measurements and vehicle trajectory – so each particle only needs to store the current map $\Theta^{(m)}$ and pose $s_t^{(m)}$ (rather than the whole trajectory $S_t^{(m)}$).

For practical purposes, when SLAM is being used to provide a pose for the rest of the vehicle control software, we usually want to turn the set particles into a single point estimate. If the posterior distribution is Gaussian, then the mean is a good estimator, but other estimators may be better if the distribution becomes non-Gaussian.

The particle filter SLAM algorithm maintains a map for each particle (the maps are initialized either blank or with a partial map of the world), and appends the map update step after the resample step in the localization sequence described above:

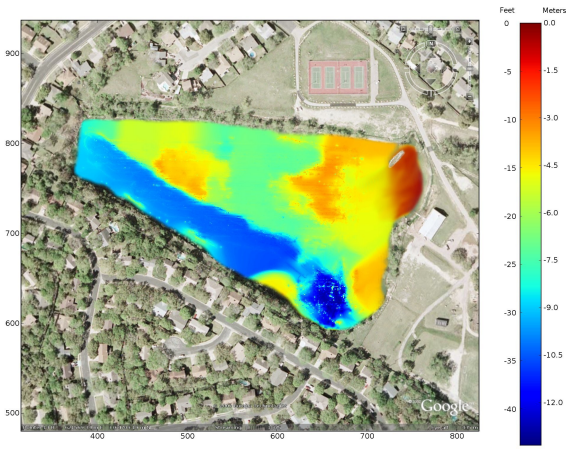

Figure 8: A map of quarry bathymetry generated using weighted k-nearest neighbor interpolation between sonar points. Coordinates are in truncated UTM northings and eastings. Very shallow regions were manually mapped with a canoe. Background image courtesy Google, ©Europa Technologies, 2007.

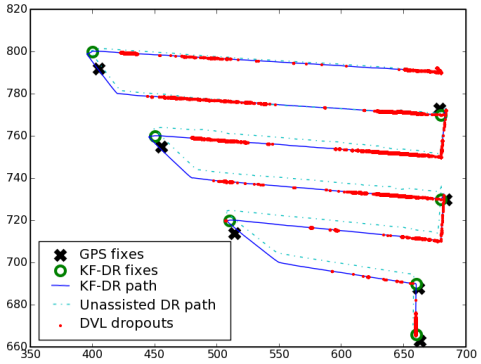

Figure 9: A comparison of dead reckoning localization performance with and without the IMU velocity Kalman filter over a raster scan mission of the quarry. Coordinates are in truncated UTM northings and eastings.

Update The measurements z are inserted into the particle maps $\Theta^{(m)}$ to update the evidence of all the voxels θ which lie in the conic sonar beam model of each measurement relative to the particle position. This is when maps must be copied and updated. We save duplicate insertions by inserting *before* copying successfully resampled particles.

6 Experiments

6.1 Hyde Park Baptist Church Quarry

We ran field tests of the DEPTHX dead reckoning system in a flooded limestone quarry in Austin, Texas (Figure 8. The quarry is a triangle roughly 200 m by 300 m, with a maximum depth in the south corner of 15 m.

Figure 9 shows the performance of the dead reckoning system during a long raster mission to map the quarry. This mission had a total path length of over a kilometer, at the end of which the positioning error was 3.16 meters. This gives a respectable dead reckoning accuracy of less than half a percent of distance traveled. It should be noted that GPS fixes used as ground truth in this experiment were taken using a hand held non-differential GPS receiver from a moving boat, so it is likely that the dead reckoning estimate is actually more accurate than the "ground truth".

The red dots in Figure 9 denote locations where the DVL failed to achieve bottom lock. During this 6300 second mission, the vehicle was without DVL measurements for a total of about 780 seconds. Some of these dropout periods were as long as sixty seconds. Given this DVL performance, it would have been impossible to achieve any reasonable dead reckoning estimate without patching in the IMU velocities as estimated by the Kalman filter.

6.2 ARL Tank Test

We first tested SLAM in a cylindrical tank (Figure 10), where the vehicle drove three cycles around a 3D box pattern (Figure 11), using dead-reckoning for real-time navigation. The box pattern was 8m on a side and 5m deep, and each cycle took about 13 minutes for a total run time of 40 minutes. The vehicle rotated $\sim 5°/s$ during ascent and descent in order to obtain better sonar coverage of the walls.

To establish the ground truth trajectory of the vehicle, we localized with 3000 particles and a manually constructed 0.25m resolution prior map of the ARL tank. The dead-reckoned trajectory drifted from the ground-truth by

Figure 10: The large wooden test tank at the University of Texas at Austin Applied Research Lab (ARL) is a cylinder 11.6 m deep and 16.8 m in diameter. Shown here is the vehicle next to the operator hut, which is in the middle of a bridge over the tank.

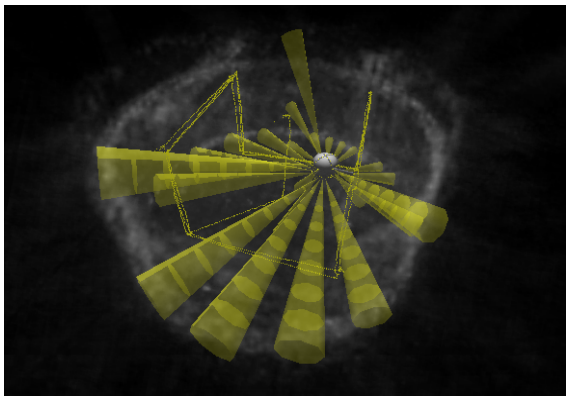

Figure 11: This figure shows the 3D trajectory of the vehicle in the ARL test tank, as well as a rendering of the vehicle and its sonar beams. The vehicle is surrounded by the cloudy evidence map constructed by SLAM, where opacity indicates occupancy.

$\sim$0.5m, which agreed with our observations during the test. We then ran SLAM using 500 particles (with no prior map), which yielded a bounded localization error of $\sim$0.1m (Figures 12 and 13).

6.3 La Pilita

La Pilita is a smaller, much more easily accesible formation that is part of the same system of cenotes as Zacatón (Figure 14). Although it exhibits much of the same morphology and biology as Zacatón, La Pilita is only about 100 m deep, and thus within the range of specialized divers if the vehicle were lost.

We successfully ran SLAM onboard the vehicle in La Pilita with 300 particles and used it to provide the real-time navigation to control the vehicle (Figure 15). Due to the lack of ground truth, we can't make any strong assertions as to the accuracy of the SLAM solution. Likewise, we can't claim any great improvement over dead reckoning, as both solutions accurately returned to the vehicle's starting location (our only ground truth point) and usually differed by less than a meter (Figure 16).

7 Conclusion

The DEPTHX vehicle is a demonstrated platform for performing exploration of fully 3D underwater environments. We have described the control and navigation systems, and demonstrated the performance of these systems using experimental results from multiple test environments, including a test tank, a flooded limestone quarry, and a cenote. Our particle filter approach to SLAM appears to work reliably, despite the low resolution, noisy, sparse, and low rate range data afforded by the pencil beam sonars – however we lack the ground truth data necessary to make definitive statements about its performance.

Thanks

We would like to thank the staff at ARL and the Hyde Park Baptist Church in Austin, Texas. We would also like to thank the other members of the DEPTHX team for their roles in bringing the DEPTHX vehicle together. In particular, we thank John Kerr and Bill Stone at Stone Aerospace and Marcus Gary at the Department of Geology at the University of Texas, Austin. This work was funded by the NASA ASTEP program, grant NNG04GC09G.

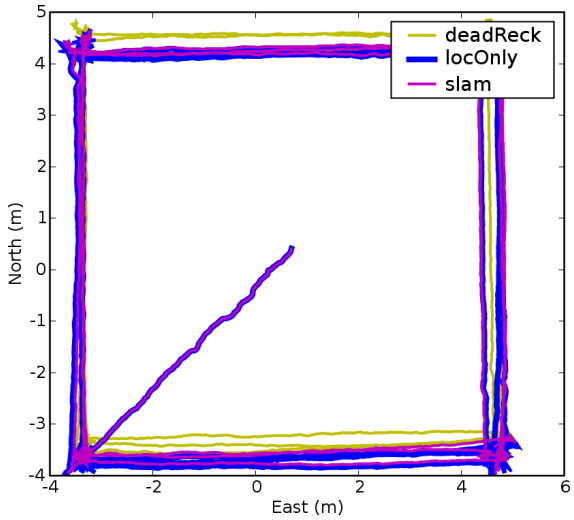

Figure 12: Planar XY view of the trajectories of the various localization solutions in the ARL test tank. The deadReck solution looks quite square as it was used to navigate during the test, but the true vehicle trajectory is shown by locOnly (localization with a prior map and 3000 particles).

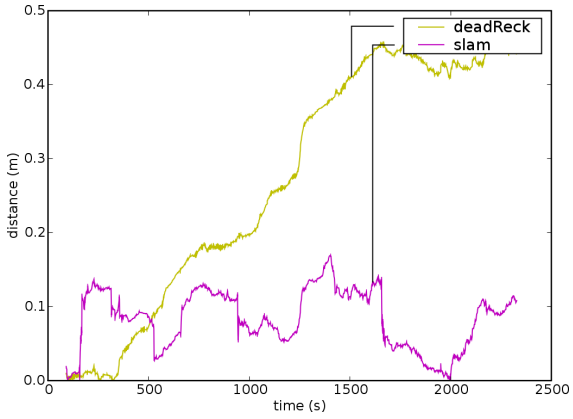

Figure 13: Distance between various localization solutions in the ARL test tank. The ground truth was established using localization with a prior map and 3000 particles (and agrees with our coarse observations during the test) – dead reckoning drifts away while SLAM error is bounded.

References

S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

J. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions on Robotics & Automation*, 17(3): 229–241, 2001.

A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proc. of the Sixteenth Conf. on Uncertainty in AI*, pages 176–183, 2000.

N. Fairfield, G. Kantor, and D. Wettergreen. Three dimensional evidence grids for SLAM in complex underwater environments. In *Proc. of the 14th Intl. Symposium of Unmanned Untethered Submersible Technology*, August 2005.

N. Fairfield, G. Kantor, and D. Wettergreen. Real-time slam with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 24:3–21, February 2007.

M. Gary. Understanding Zacatón: Exploration and initial interpretation of the world's deepest known phreatic sinkhole and related karst features, southern Tamaulipas, Mexico. *Karst Frontiers, Karst Waters Institute Special Publication*, 7:141–145, 2002.

M. B. Larsen. High performance doppler-inertial navigation experimental results. In *Proc. of IEEE/MTS OCEANS*, pages 1449–1456, 2000.

J. Leonard, A. Bennett, C. Smith, and H. Feder. Autonomous underwater vehicle navigation. Technical report, MIT Marine Robotics Laboratory, 1998.

M. Martin and H. Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 1996.

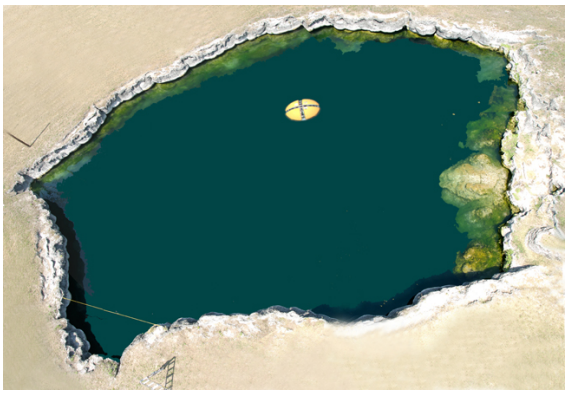

Figure 14: La Pilita aerial view, with the DEPTHX vehicle at the surface. The mouth of the cenote is only about 30 m across although it opens up considerably at depth. Courtesy Robyn Gary and Jason Sahl.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. of the AAAI National Conference on Artificial Intelligence*, pages 593–598, 2002.

K. Murphy. Bayesian map learning in dynamic environments. In *Neural Information Processing Systems*, pages 1015–1021, 1999.

C. Roman. *Self Consistent Bathymetric Mapping from Robotic Vehicles in the Deep Ocean*. PhD thesis, Massachusetts Institute of Technology & Woods Hole Oceanographic Institution, May 2005.

L. Whitcomb. Underwater robotics: Out of the research laboratory and into the field. In *IEEE 2000 International Conference on Robotics and Automation*, May 2000.

S. Williams and I. Mahon. Simultaneous localisation and mapping on the great barrier reef. In *Proc. of IEEE Intl. Conf. on Robotics and Automation*, volume 2, pages 1771–1776, April 26-May 1 2004.

S.B. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *Proc of IEEE Conf. on Robotics and Automation*, May 2000.

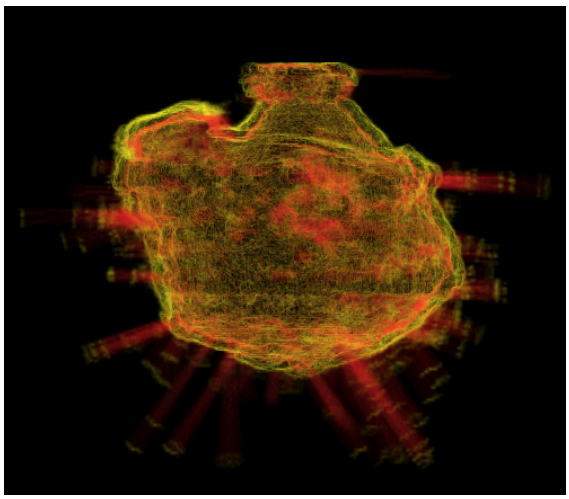

Figure 15: An east-facing orthogonal view of a SLAM evidence grid of La Pilita. The maximum depth of La Pilita is 117 m, and its width is about 100 m, while the narrow neck that opens to the surface is only about 30 m wide. In this figure, yellow indicates an occupancy isosurface, and red indicates a vacancy isosurface. Individual beams, caused by spurious sonar measurements, can be seen projecting through the surface of the cenote.

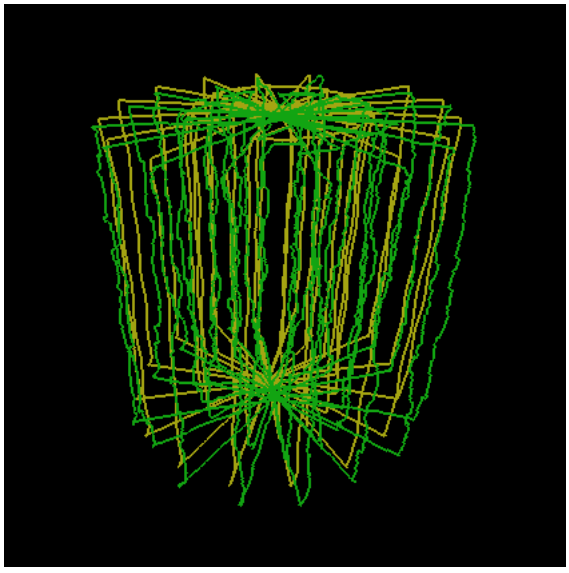

Figure 16: A perspective view of the nested star path that the vehicle followed to map La Pilita, the dead-reckoned trajectory in yellow, and the SLAM trajectory in green.